DEVELOPMENT OF LEVEL CONTROLLERS BASED ON FUZZY LOGIC

A. Cabrera[1], R. Senhadji[2], S. Sánchez-Solano[2], A. Barriga[2],
C. J. Jiménez[2], O. Llanes[1]

[1] Dpto. de Automática y Computación, Facultad de Ingeniería Eléctrica,
Universidad Politécnica de la Habana, (ISPJAE), calle 127 s/n, apartado 6028,
Marianao, Ciudad de la Habana, Cuba

[2] Instituto de Microelectrónica de Sevilla - Centro Nacional de Microelectrónica
Avda. Reina Mercedes s/n, (Edif. CICA)
E-41012, Sevilla, Spain

Proc. 1st International ICSC-NAISO Congress on Neuro-Fuzzy Technologies (NF2002), pp. 81, La Habana, January 16-19, 2002.

Development of Level Controllers Based on Fuzzy Logic*

Alejandro Cabrera[1], Raouf Senhadji[2], Santiago Sánchez-Solano[2], Ángel Barriga[2], Carlos J. Jiménez[2], Orestes Llanes[1]

[1] Dpto. de Automática y Computación, Facultad de Ingeniería Eléctrica, Universidad Politécnica de la Habana, (ISPJAE), calle 127 s/n, apartado 6028, Marianao, Ciudad de la Habana, Cuba
[2] Instituto de Microelectrónica de Sevilla, Centro Nacional de Microelectrónica, Avda. Reina Mercedes s/n, Edificio CICA, E-41012, Sevilla, Spain

Abstract.

This paper describes the development of different kinds of level controllers based on fuzzy logic. Designs and implementations were carried out using tools from *xfuzzy*, a development environment that eases the different stages in the design of fuzzy inference systems. Special emphasis has been put in the on-line verification of the controller over a physical plant by means of *xflab* tool. Different approaches of the knowledge base were tested using *xflab*. Then, the results were analyzed and selected those that gave the better performance. Controllers implementations with different number of bit for input / output resolution were also carried out and analyzed. The results provide a base for the incoming development of a hardware fuzzy logic controller by means of specific hardware or by an embedded codesign system.

I. Introduction.

A lot of applications using level controllers have been reported in the literature. Many of them employ traditional like-PID procedures for automation. However, very often the process to control is a nonlinear one whose mathematical model is too difficult to obtain or it exists but is very complex to encode. So, the use of traditional controllers in these processes has its drawbacks. Fuzzy logic has become one of the most successful of today's technologies for the development of control systems [1,2]. Fuzzy controllers are simpler to describe and easier to implement than traditional controllers.

There are reports about level controllers based on fuzzy logic [3]. Also, there are many CAD tools for the development of fuzzy logic controllers [4]. Most of them can simulate the controller's operation by combining the fuzzy system model with a simplified model of the real process. Some of these tools have also the capabilities to include the real process in the control loop in order to carry out the on-line verification of the whole system (controller plus process) [5].

This paper describes the development of different kinds of level controllers based on fuzzy logic. Designs and implementations were carried out using different tools from the *xfuzzy* development environment. The paper is organized as follows. Section II shows the description of the physical plant where the controllers were developed and applied. The *xfuzzy* development environment and also its on line verification tool, *xflab,* are described in section III. Section IV shows different controller's implementations, presents the test realized and the obtained results. Section V shows the summary of the present communication.

II. System Description.

Figure 1 shows the physical plant where the controllers were applied. The system is composed of an electrically controlled water pump and two cylindrical tanks (120 cm high and 20 cm wide), each one with an electronic valve for liquid injection and a manually controlled output valve for liquid discharge. The liquid level is measured through a pressure sensor located at the bottom of the tank. The water pump and the injection valves are voltage-controlled devices (0 – 10 volt) whereas the pressure sensors have current signal outputs (4 – 20 mA).

Three different types of level-controlled systems were tested with this plant. Two of them are composed by only one tank and the difference between both is the control

* This work has been carried out in collaboration with SUR A&C S.L.

element: the electronic valve or the water pump. In both cases, the other device (the pump or electronic valve) remains at a fixed position.

Figure 1. Two-tank level system provided by SUR A&C.

These one-tank systems are functionally equivalents with fuzzy control actions based on equivalent rules. For instance, if the current liquid level is far away from the target position, the rule:

"if *error* is HIGH then OPEN WIDELY the valve"

is equivalent to the rule:

"if error is HIGH then PUMP A LOT OF liquid".

On the other hand, if the liquid level is at target position, the rule:

"if error is ZERO then DO NOT OPEN the valve"

is equivalent to the rule:

"if error is ZERO then DO NOT PUMP liquid".

So, the same fuzzy controller can be applied to both one-tank systems.

The third type of system is the whole plant where the controller must act over the motor pump and both electronic valves at the same time. The goals in the three systems are to maintain the level in each tank as near as possible to the target control positions with the lower overshoot and with the lesser transient time.

The control strategy carried out in the three systems was the fuzzy equivalent of an incremental PD controller (equation 1), where *error* and *Δerror* are defined by equation 2 and 3.

$$\Delta out = f(error, \Delta error) \quad (1)$$

$$error = L_T - L_C \quad (2)$$

$$\Delta error = error(t-1) - error(t) \quad (3)$$

where L_T and L_C are the target and current levels respectively, and error(t-1) and error(t) are error values at the previous sampling time and at the present one, respectively.

Therefore, the fuzzy controller for one-tank systems has two inputs (the error and its change) and its output is the change of the valve aperture or the change of the pump motor speed. Two independent fuzzy controllers were used for the whole plant, each one acting upon one electronic valve, whereas the pump was controlled by the bounded sum of the output of both controllers.

III. The *Xfuzzy* Environment and the On-Line Verification Tool *Xflab*.

Designs and implementations were carried out using different tools from *xfuzzy* [4,6], a powerful fuzzy logic development environment that eases the description, verification, and synthesis of fuzzy controllers.

This environment allows, among other facilities, the specification of a fuzzy system, its simulation, its on-line verification working over the real process, and its implementation, either by means of software or hardware. The hardware synthesis tools, as well as the on-line verification tool, are the main facts that distinguish *xfuzzy* from other similar environments. Version 2.1 of *xfuzzy* runs on any Unix compatible or Microsoft Windows operating system.

The design flow using *xfuzzy* is shown in Figure 2. All the tools integrated on *xfuzzy* use the fuzzy systems description language XFL [7]. An XFL specification contains information concerning the input / output membership

functions, the system rule base, the inference mechanism and the defuzzification method. Modifications on these parameters leads to different system behavior, so, tools to evaluate the influence of each one are required.

It is possible to approximate to the system's behavior starting from an XFL specification and using the simulation tool *xfsim*, incorporated in *xfuzzy*. Simulation can be performed with the aid of a model of the plant written in C language. But this model cannot ever reflect the real operation of the physical plant. However, it is the connection from the controller under development to the real process the one that provides the better information and also the one that allows the tuning of parameters. The tool in charge of this task in the *xfuzzy* environment is *xflab* [5].

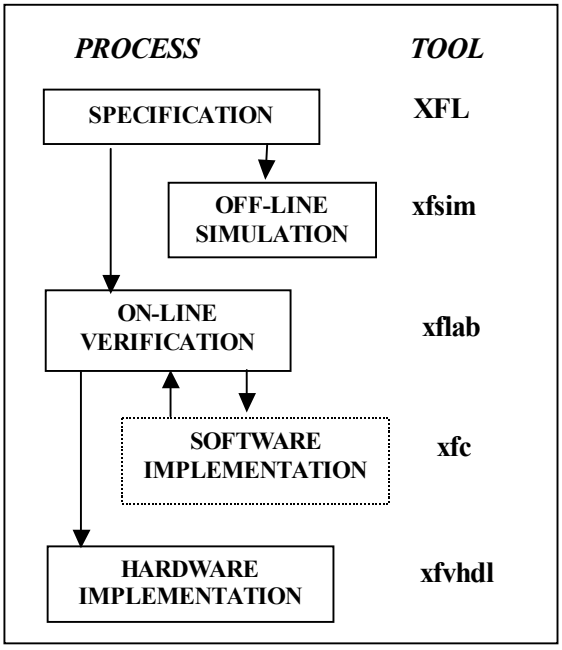

Figure 2. Design flow using *xfuzzy*.

Even more, *xflab* allows the fully development of a software based fuzzy logic controller, by means of a personal computer and a PC data acquisition card with analog and digital inputs / outputs. Using this tool, the model of the plant used in standard simulation is replaced by a function that monitors the real plant and sends this information to the fuzzy controller. Once processed, the function takes the controller's results and acts over the physical plant.

The current version of *xflab* uses National Instruments Lab PC or compatible data acquisition boards. Others cards can be employed writing their corresponding drivers. National Instruments PCI-1200 board was used in this work. It has eight 12 bits analog inputs, two 12 bits analog outputs and 24 digital input / output lines. An external card was also needed in order to couple the sensors and actuators to the PCI-1200 card.

The data acquisition board must be configured in *xflab* by defining the base I/O address and assigning analog and digital signals. This configuration is stored in a *.in* file. Two other files are needed in order to run *xflab*: a *.xfl* file that contains the fuzzy controller's description, and another one (*.ctl* file) needed for pre- and post-processing conditions (i.e. filtering and linealization) and also for defining control parameters (i.e. sampling rate, number of iterations, and signals to be monitored).

Once defined the input / output configuration for the real process interface card and programmed the pre- and post-processing conditions of the controller by means of C code, *xflab* calls for the execution of *xfc*, another tool of *xfuzzy*, in charge of to obtain the C code corresponding to the fuzzy inference system described by XFL. All these codes are compiled and linked together by *xflab*. Thus, an executable file that corresponds to the software fuzzy logic controller is achieved. Then, the execution of the whole control process is started. First, the plant is monitored by reading the input ports of the PCI-1200 card and applying the data pre-processing routines. Afterwards, data are sent to the fuzzy inference mechanism whose outputs are post-processed and sent to the real plant through the output ports of the card. This process goes on until the end condition specified by the user is reached. During execution, *xflab* stores in different files the values of monitored signals for a later processing.

Note that it is very easy to change controller's specification (modifying the specification file) and to record the system performance. So, different conditions can be tested in order to obtain the better approach for a future hardware implementation of the controller.

IV. Fuzzy Logic Controllers Design and Analysis Results.

The *xfuzzy* environment has different tools that lead to hardware implementation of fuzzy inference system. One of them is *xfvhdl,* which translates an XFL specification

into a VHDL description using a specific architecture [4]. This architecture is based on active rules processing, overlapping degree of membership function limited to two, and simplified defuzzification methods. In order to achieve a controller that could be further implemented by this specific hardware architecture, some conditions in the controllers were defined. Thus, the defuzzification method was Fuzzy Mean and the number of membership functions for inputs was limited to three, and five singletons for output (see Figure 3).

The input and output universes of discourse were estimated based on the analysis of the dynamic behavior of the real process. Asymmetries of *error* and *Δerror* membership functions are due to the difference between the input and output liquid flow. Figure 3 also shows the system's rule base.

Although there are a lot of parameters that can be modified in a fuzzy control system, the work was focused on those that have the greater influence over an incoming hardware implementation with the specific architecture mentioned above. Thus, different adjusts to membership functions were carried out with *xflab* and their influences over the controller results were analyzed, selecting those that gave the better results. Three different sets of membership function were considered for each input and output. This led to a total of 27 implementations that were tested over the real plant. These sets were obtained by narrowing or stretching the central membership functions, thus making greater or lesser the influence of the controller's inputs and output.

Table 1 shows the significant values of this function for all implementations. Only modified points, extreme values of central membership functions, are shown. All the tests were made under the same conditions: a sampling time of 500 ms and 2000 samples were taken, with changes of the reference level (50, 80, 40 y 50 cm), controlled by software, at specific times. The perturbation (i.e., opening or closing the output valve) was also similar in all tests, although these were manually controlled operations, so exactly the same conditions were impossible to obtain.

Five different variables were registered by *xflab* in each test: *error*, *Δerror, Δout,* and also the current level and the total voltage applied to the actuator. Then, the data were processed and a group of control system quality parameters was obtained. These parameters were the rise and setting time, the overshoot, the maximum value of

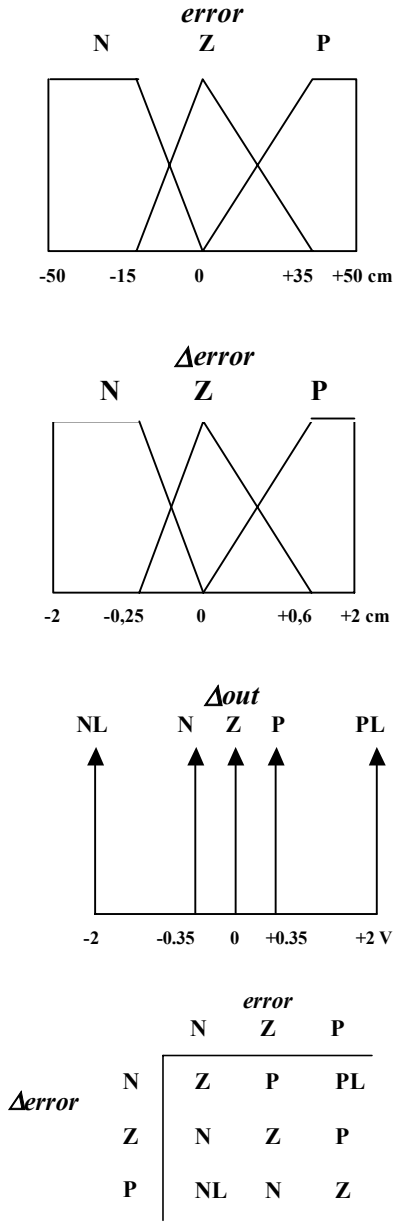

Figure 3. Input / Output membership functions and rule base

error variable and the mean quadratic error (MQE). Table 1 only shows MQE values for simplicity. The analysis of these parameters led to select the better specification (test number five). These are the values of membership functions shown in Figure 3.

These tests were carried out in one-tank systems and the knowledge base that led to the best results applied to the whole two-tank system, obtaining very good results. Figure 4 shows the liquid level versus samples in the two-tank system where the good performance of the controller is obvious. The overshoot that appears at sample 500 was due to a very drastic change in the output valve when the level in the tank was very near to the reference level. Similar conditions led to undershoot also shown in Figure 4. However, observe the short time that the controller needs in order to reach the reference level again.

.

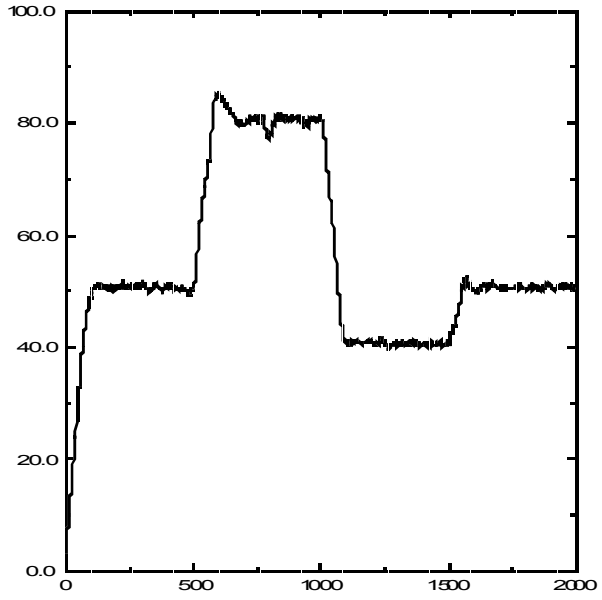

Figure 4. Tank 2 liquid level vs. samples (each sample corresponds to 500 ms)

With the incoming hardware implementation mentioned earlier in mind, controllers implementations with different number of bit for input / output resolution were also carried out and analyzed. These test were quite important due to the limited precision of hardware in order to obtain the best cost/performance ratio. Implementations with 12, 8, 6, 5 and 4 bits were evaluated. The results show that 6 or more bits systems have very similar performance, a very important conclusion for hardware implementation. The results with 4 bits show an error greater than 10%, so this reduction is not recommended. Controllers with five bits report acceptable results with an error less than 10 %.

The results provide a base for the incoming development of a hardware fuzzy logic controller by means of specific hardware or by an embedded codesign system

V. Summary.

The development of level controllers based on fuzzy logic has been shown. They were carried out with the fuzzy logic design environment *xfuzzy*. Inference systems were implemented in software but with constraints for an incoming hardware implementation based on a specific architecture. Controllers were developed in an on-line verification scheme with the aid of *xflab* tool. Different tests over controllers in order to obtain the better performance were made. These tests modified the membership functions and the input / output resolution.

Although this paper is limited to the use of *xflab* for the controller development and parameters adjust, an inference system hardware implementation, based on the previous software controller results, was later carried out in an embedded codesign environment composed by microcontrollers and FPGAs. The pretty good obtained results evidence the powerfulness of *xflab* and also that a complete codesign methodology can be developed based on *xfuzzy* tools.

References.

[1] Passino, K., Yorkovich, S.,"Fuzzy Control", Addison Wesley, 1998

[2] Reznik, L., "Fuzzy Controllers", Victoria Univ. Of Technology, Ed. Newness, 1997

[3] Heckenthaler, T. and Engell, S.: "Aproximately Time-Optimal Fuzzy Control of a Two-Tank System", IEEE-Control Systems Magazine, vol 14, No. 3, pp.24-30, Jun 1994

[4] Baturone, I., Barriga, A., Sánchez-Solano, S., Jiménez, C.J., and López, D.: "Microelectronic Design of Fuzzy Logic-Based Systems", CRC Press, 2000

[5] Senhadji, R., Sánchez-Solano, S., López, D.R. and Barriga, A.: "Xflab: An On-line Verification Tool for Fuzzy Controllers", Proc. International Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems", Vol. 1, pp. 44-49, Madrid, Jul 2000.

[6] López, D.R., Jiménez, C.J., Baturone, I., Barriga, A. and Sánchez-Solano, S.: "Xfuzzy: A Design Environment for Fuzzy Systems", Proc. seventh IEEE International Conference on Fuzzy Systems, pp. 1060-1065, Anchorage, May 1998

[7] López, D.R., Moreno, F.J., Barriga, A. and Sánchez-Solano, S.: "XFL: A Language for the Definition of Fuzzy Systems", Proc. sixth IEEE International Conference on Fuzzy Systems, Vol. 3, pp. 1581-1591, Barcelona, Jul 1997

Table 1. Test conditions and results.

Test	*error*	*Δerror*	*Δout*	*MQE1*	*MQE2*	*MQE3*	*MQE4*
T1	**-15, +35**	**-0.15, +0.4**	**-0.2, +0.2**	**4,80**	**7,11**	**5,35**	**3,89**
T2	**-15, +35**	**-0.15, +0.4**	**-0.35, +0.35**	**2,51**	**5,98**	**4,92**	**1,79**
T3	**-15, +35**	**-0.15, +0.4**	**-0.5, +0.5**	**2,62**	**4,39**	**---**	**---**
T4	**-15, +35**	**-0.25, +0.6**	**-0.2, +0.2**	**4,85**	**6,12**	**5,85**	**3,90**
T5	**-15, +35**	**-0.25, +0.6**	**-0.35, +0.35**	**2,89**	**2,69**	**1,94**	**3,31**
T6	**-15, +35**	**-0.25, +0.6**	**-0.5, +0.5**	**3,42**	**4,05**	**2,19**	**3,18**
T7	**-15, +35**	**-0.35, +0.8**	**-0.2, +0.2**	**4,23**	**5,21**	**4,25**	**3,29**
T8	**-15, +35**	**-0.35, +0.8**	**-0.35, +0.35**	**4,92**	**4,15**	**3,06**	**2,88**
T9	**-15, +35**	**-0.35, +0.8**	**-0.5, +0.5**	**3,56**	**6,65**	**3,30**	**5,30**
T10	**-25, +45**	**-0.15, +0.4**	**-0.2, +0.2**	**7,24**	**9,24**	**7,50**	**2,38**
T11	**-25, +45**	**-0.15, +0.4**	**-0.35, +0.35**	**5,93**	**5,25**	**3,81**	**3,74**
T12	**-25, +45**	**-0.15, +0.4**	**-0.5, +0.5**	**5,61**	**3,86**	**2,50**	**4,54**
T13	**-25, +45**	**-0.25, +0.6**	**-0.2, +0.2**	**5,30**	**---**	**---**	**---**
T14	**-25, +45**	**-0.25, +0.6**	**-0.35, +0.35**	**3,31**	**8,20**	**4,13**	**3,72**
T15	**-25, +45**	**-0.25, +0.6**	**-0.5, +0.5**	**4,02**	**4,15**	**2,26**	**7,24**
T16	**-25, +45**	**-0.35, +0.8**	**-0.2, +0.2**	**7,12**	**---**	**---**	**---**
T17	**-25, +45**	**-0.35, +0.8**	**-0.35, +0.35**	**4,48**	**6,47**	**1,23**	**3,24**
T18	**-25, +45**	**-0.35, +0.8**	**-0.5, +0.5**	**4,32**	**11,29**	**5,11**	**4,60**
T19	**-35, +55**	**-0.15, +0.4**	**-0.2, +0.2**	**6,79**	**10,17**	**2,64**	**3,49**
T20	**-35, +55**	**-0.15, +0.4**	**-0.35, +0.35**	**7,70**	**6,95**	**2,34**	**2,35**
T21	**-35, +55**	**-0.15, +0.4**	**-0.5, +0.5**	**4,37**	**3,81**	**3,18**	**5,35**
T22	**-35, +55**	**-0.25, +0.6**	**-0.2, +0.2**	**10,36**	**13,16**	**4,25**	**2,61**
T23	**-35, +55**	**-0.25, +0.6**	**-0.35, +0.35**	**2,55**	**5,77**	**2,47**	**3,71**
T24	**-35, +55**	**-0.25, +0.6**	**-0.5, +0.5**	**8,05**	**5,20**	**6,03**	**3,08**
T25	**-35, +55**	**-0.35, +0.8**	**-0.2, +0.2**	**9,67**	**8,45**	**4,30**	**4,14**
T26	**-35, +55**	**-0.35, +0.8**	**-0.35, +0.35**	**4,73**	**6,96**	**4,26**	**2,51**
T27	**-35, +55**	**-0.35, +0.8**	**-0.5, +0.5**	**3,99**	**9,46**	**3,78**	**7,86**